Problem

假设有一个游戏，有 n 关，m 种怪物，第 i 种怪物会出现在第 p_i 到第 q_i 关（$p_i \le q_i$），仅在第一次见到第 i 种怪物时会损失 a_i 点生命值，每通过一关（这意味着见到这一关会出现的所有种类的怪物）则会获得 b_i 点生命值。考虑集合 $S_{lr} = \{x|x \in \mathbb{Z}, l \le x \le r\}$，定义：

$$f(l,r) = \max_{T \subseteq S_{lr}, T \neq \emptyset} \{\sum_{i \in T} b_i - \sum_{[p_i,q_i] \cap T \neq \emptyset} a_i\}$$

即 $f(l,r)$ 为：从 S_{lr} 中选出一个非空子集 T，只游玩 T 中的关卡，所有方案中生命值变化量（所有获得的生命值减去所有损失的生命值）的最大值。q 次询问，每次给出 l, r，求 $f(l,r)$ 的值。

$1 \le n, m, q \le 5 \times 10^5, 1 \le a_i, b_i \le 10^9$。

Editorial

令：

$$g(l,r) = \min_{T \subseteq S_{lr-1}} \{ \sum_{[p_i,q_i] \cap (T \cup \{r\}) \neq \emptyset} a_i - \sum_{i \in T \cup \{r\}} b_i\}$$

则 $f(l,r) = -\min_{l \le i \le r}\{g(l,i)\}$。

考虑如何计算 $g(l,r)$。我们分两种情况讨论：固定 l，由 $g(*,r)$ 推 $g(*,r+1)$；固定 r，由 $g(l,*)$ 推 $g(l+1,*)$。

固定 l

记 $cnt(i,j) = \sum_{i < p_k \le j, q_k \ge j} a_k$，显然有转移：

$$g(l,r) = \min\{cnt(0,r), \min_{l \le i < r}\{g(l,i) + cnt(i,r)\}\} - b_r$$

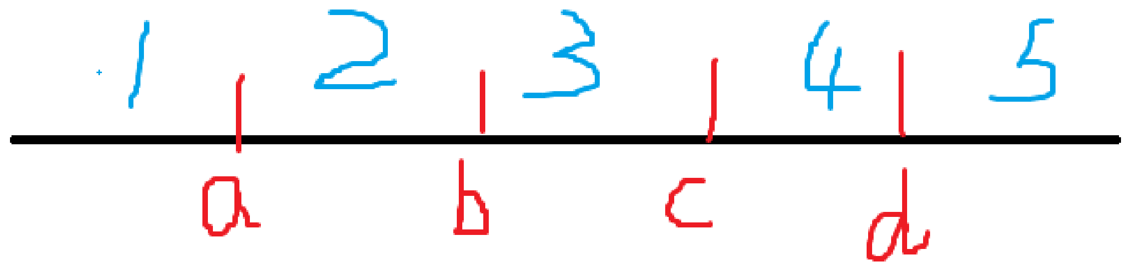
其中，$cnt(0,r)$ 是易于维护的，而 $g(l,i) + cnt(i,r)$ 可以使用线段树维护，因此可以用 $O(nlogn)$ 的时间复杂度算出 $f(1,*)$。

固定 r

这部分主要考虑当 l 变为 $l+1$ 时，$g(l+1,r)$ 与 $g(l,r)$ 的关系。

记 i 的最优转移点为 $bp(i)$（若 i 的转移取 $cnt(0,i)$ 一项，则认为 $bp(i) = 0$），$bp^k(i) = \begin{cases} bp(bp^{k-1}(i)) & k > 0 \\ i & k = 0 \end{cases}$，$g(l,i)$ 所对应的选择方案的集合为 $BP(i)$。注意到转移方程的实际意义是枚举 $BP(r)$ 中的次大元素，因此我们有：$BP(i) = (\cup_{k \ge 0}\{bp^k(i)\}) \setminus \{0\}$。

考虑前一情况转移中的 $\min_{l \le i < r}\{g(l,i) + cnt(i,r)\}$ 的部分，令 $a < b < c < d$，考虑 $g(l,c)$ 和 $g(l,d)$ 的最优转移点：有两种情况，$g(l,c)$ 的最优转移点是 $g(l,a)$（即 $bp(c) = a$），$g(l,d)$ 的最优转移点是 $g(l,b)$（即 $bp(d) = b$）；或者相反。分别计算这两种情况中 $g(l,c) + g(l,d)$ 的值：



记 $c(i,j)$ 表示左端点在上图编号为 i 的区间中，右端点在上图编号为 j 的区间中的所有 a_i 的和，那么：

转移区间交叉的情况（即 $g(l,a)\to g(l,c), g(l,b)\to g(l,d)$）

$$\begin{aligned} g(l,c)+g(l,d) = &\ (g(l,a)+c(2,4)+c(2,5)+c(3,4)+c(3,5)-b_c) \\ &+(g(l,b)+c(3,5)+c(4,5)-b_d) \end{aligned}$$

转移区间包含的情况（即 $g(l,a)\to g(l,d), g(l,b)\to g(l,c)$）

$$\begin{aligned} g(l,c)+g(l,d) = &\ (g(l,b)+c(3,4)+c(3,5)-b_c) \\ &+(g(l,a)+c(2,5)+c(3,5)+c(4,5)-b_d) \end{aligned}$$

观察上述两种情况，可以发现前一种情况相比后一种情况，多一项 $c(2,4)$。若令
$w(i,j)=g(l,i)+cnt(i,j)-b_j$，那么：

$$w(a,c)+w(b,d)\geq w(a,d)+w(b,c)$$

恒成立。（注：反向的四边形不等式，熟悉dp优化的选手想到这一步是很自然的）

固定 c,d，a 和 b 之间的关系是我们关心的，接下来对此展开讨论。

假设 $a<b<c=r$，那么根据最优化条件，$w(a,c)<w(b,c)$ 且 $w(b,d)<w(a,d)$，即
$w(a,c)+w(b,d)<w(a,d)+w(b,c)$，而这是与上述不等式是矛盾的，因此假设不成立。于是我们得出结论：对于任意的 c,d，若 $c<d$，则要么 $bp(d)\leq bp(c)$，要么 $c\leq bp(d)<d$。

上述结论有一个重要推论：$bp(i+1)\subseteq BP(i), BP(i+1)\setminus\{i+1\}\subseteq BP(i)$，也即存在整数 n，满足 $bp(i+1)=bp^n(i)$，$BP(i+1)=(\cup_{k\geq n}\{bp^k(i)\})\setminus\{0\}\cup\{i+1\}$。证明如下：

令 $c=i,d=i+1$，上述结论可以改写为：要么 $bp(i+1)=i$，要么 $bp(i+1)\leq bp(i)$。进一步的，令 $c=bp(i),d=i+1$，注意到 $bp(i)=c<bp(i+1)<i$ 是不可能的，因此要么 $bp(i+1)=bp(i)$，要么 $bp(i+1)\leq bp^2(i)$。这一过程可以一直持续下去，由数学归纳，可知 $bp(i+1)\subseteq BP(i)$。

上述推论让我们对 $BP(r)$ 的构成有了更直观的认识。显然，假若对某一 r_0，$l\notin BP(r_0)$，那么：1. 任意 $r>r_0$，$l\notin BP(r)$；2. 总存在一个最小的 $r_0=r_m$，使得当 $r<r_m$ 时，$l\in BP(r)$，且 $BP(r_m)=\{r_m\}$。

显然，当 l 变为 $l+1$ 时，所有 $r\geq r_m$ 的 $g(l,r)$ 的值是不变的，接下来考虑 $r<r_m$ 的 $g(l,r)$ 的变化情况，为此，我们考察 $BP(r)$ 的变化情况，这里，我们假定 r_m 足够大（否则是平凡的）。记 $BP'(r)$ 为 $g(l+1,r)$ 对应的选择方案的集合。

显然有 $BP'(l+1)=\{l+1\}$，而 $BP(l+1)=\{l,l+1\}$，因此
$g(l+1,l+1)=g(l,l+1)+b_l-\sum_{q_i=l}a_i$，令 $\Delta_l=b_l-\sum_{q_i=l}a_i$，可以改写为
$g(l+1,l+1)=g(l,l+1)+\Delta_l$，这里 Δ_l 是一个“基本量”——显而易见的，任意满足 $l+1\in BP(r)$ 的 r，总存在一种可能性，满足 $BP'(r)=BP(r)\setminus\{l\}$，从而 $g(l+1,r)=g(l,r)+\Delta_l$，即不等式 $g(l+1,r)\leq g(l,r)+\Delta_l$ 总是满足的。更一般的，我们可以定义 $\Delta_x=cnt(0,x)-b_x-g(l,x)$，这一重要变量将在后文反复出现。

注意到 $BP(i+1) \setminus \{i+1\} \subseteq BP(i)$，因此存在一个最小的 $r = r_1$，满足当 $r < r_1$ 时 $l+1 \in BP(r)$；$r \geq r_1$ 时 $l+1 \notin BP(r)$。对于 r_1，我们知道 $BP(r_1) = \{l, r_1\}$，显然 $BP'(r_1) = \{r_1\}$，因此 $g(l+1, r_1) = cnt(0, r_1) - b_{r_1} = g(l, r_1) + \Delta_{r_1}$。接下来考虑 $r < r_1$ 的部分。这里，根据上述讨论，有 $g(l+1, r) \leq g(l, r) + \Delta_l$，我们接下来证明一个引理：

引理1：存在一个最小的 $r = r_2$，满足当 $r < r_2$ 时 $l+1 \in BP'(r)$ 即 $g(l+1, r) = g(l, r) + \Delta_l$；$r \geq r_2$ 时 $l+1 \notin BP'(r)$ 即 $g(l+1, r) < g(l, r) + \Delta_l$。

证明：设 $r_2' = \inf\{r | l+1 \notin BP'(r)\}$ 表示最小的满足 $l+1 \notin BP'(r)$ 的 r，设 $BP(r_2') = \{s_1, s_2, s_3, \dots, s_{k-1}, s_k\}$，其中 $s_1 = l, s_2 = l+1, s_k = r_2'$。注意到 $BP'(r) \setminus BP(r) = \emptyset$（这是因为，假若 $BP'(r) \setminus BP(r) \neq \emptyset$，那么选择 $BP'(r) \cup \{l\}$ 和 $BP'(r)$ 中的较优的方案将会得到一个更优的 $g(l, r)$，这与最优化条件矛盾），同时 $BP'(r_2) \subseteq BP'(s_{k-1}) = BP(r_2) \setminus \{l, r_2\}$，因此存在一个 $n < k$，满足 $BP'(r_2) = \{s_2, s_3, \dots, s_{n-1}, s_n, s_k\}$。注意到假若 $n \geq 2$，那么选择 $BP'(r_2) \cup \{l\}$ 同样得到一个更优的 $g(l, r)$，因此 $n = 1$，即 $BP'(r_2) = \{r_2\}$，即 $l+1 \notin BP'(r_2)$。又由 $BP(i+1) \setminus \{i+1\} \subseteq BP(i)$，$l+1 \notin BP'(r)$ 对所有 $r > r_2$ 的 r 也成立，引理得证。

注意到此时 $BP'(r_2) = \{r_2\}$，从而 $g(l+1, r_2) = g(l, r_2) + \Delta_{r_2}$。又由 $g(l+1, r_2) < g(l, r_2) + \Delta_l$，得 $\Delta_{r_2} < \Delta_l$。进一步的，当 $r < r_2$ 时，$\Delta_r \geq \Delta_l$ 总是成立。因此可以通过查找第一个满足 $\Delta_r < \Delta_l$ 的 r 的方式找出 r_2。

我们发现，当 $r_2 < r < r_1$ 时，可以规约到 $r \leq r_2$ 的情况，从而有下述引理：

引理2：当 $r < r_1$ 时，设序列 $p_1, p_2, \dots, p_k$ 满足 $l = p_1 < p_2 < \cdots p_k$，$\forall i > 2, p_{i-1} < j < p_i$，均有 $\Delta_{p_i} < \Delta_{p_{i-1}} < \Delta_j$，且 $\forall j \in (p_k, r_1), \Delta_{p_k} < \Delta_j$，有：令 $M = \sup\{x | p_x \leq r\}$，那么 $g(l+1, r) = g(l, r) - \Delta_M$。特别地，记 $p_k = r_3$。

接下来我们考虑 $r \geq r_1$ 的情况，我们希望将引理2中 $r < r_1$ 的限制去掉，即证明下述引理：

引理3：$\Delta_{r_1} \leq \Delta_{r_3}$。

证明：考虑一种选取方案 $S = \{l, r_3\}$，其对应的权值为 $cnt(0, r_3) - b_{r_3} + \sum_{p_i \leq l, l \leq q_i < r_3} a_i - b_l$，令 $\Delta_{r_3}' = b_l - \sum_{p_i \leq l, l \leq q_i < r_3} a_i$，那么可以改写为 $cnt(0, r_3) - b_{r_3} - \Delta_{r_3}'$。根据最优化条件，这一权值不小于 $g(l, r_3)$，即 $cnt(0, r_3) - b_{r_3} - \Delta_{r_3}' \geq g(l, r_3) = cnt(0, r_3) - b_x - \Delta_{r_3}$，从而 $\Delta_{r_3} \geq \Delta_{r_3}'$。注意到 $r_1 > r_3$，因此有 $\Delta_{r_1} = b_l - \sum_{p_i \leq l, l \leq q_i < r_1} a_i \leq b_l - \sum_{p_i \leq l, l \leq q_i < r_3} a_i = \Delta_{r_3}' \leq \Delta_{r_3}$，因此引理得证。

利用引理3，我们可以将引理2推广为：

- 设序列 $p_1, p_2, \dots, p_k$ 满足 $l = p_1 < p_2 < \cdots p_k$，$\forall i > 2, p_{i-1} < j < p_i$，均有 $\Delta_{p_i} < \Delta_{p_{i-1}} < \Delta_j$，且 $\forall j \in (p_k, n), \Delta_{p_k} < \Delta_j$，有：令 $M = \sup\{x | p_x \leq r\}$，那么 $g(l+1, r) = g(l, r) - \Delta_M$。

这样，我们完整的得到了 $g(l, *)$ 推 $g(l+1, *)$ 的公式。注意到对于任意元素 r，其成为某一 p_i 就意味着 $BP'(r) = \{r\}$，因此在 l 从 1 迭代到 n 的过程中会且仅会出现一次，从而暴力找 p 的过程最多出现 n 次。综上所述，我们可以在 $O(nlogn)$ 的时间内算出任意的 $g(l, r)$。